

EV355228524

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Operating System Deployment Methods and Systems

Inventor(s):
Shantanu Sardesai
Shafqat Khan
Deepak Kumar
Gopal Parupudi
Vinay Deo

ATTORNEY'S DOCKET NO. ms1-1613us

RELATED APPLICATION

This application stems from and claims priority to U.S. Provisional Application Serial No. 60/455,830, filed on March 18, 2003, the disclosure of which is incorporated by reference herein.

TECHNICAL FIELD

This invention pertains to operating system deployment methods and systems.

BACKGROUND

Deploying and supporting operating systems is probably one of the most complex tasks that an organization's information technology department faces. Deploying a new operating system (whether the new operating system constitutes the old operating system with a new set of applications, or a *de facto* new operating system) typically involves, for each machine in the organization, removing the old operating system and installing the new operating system, along with the software mix that runs on top of the new operating system. In a large organization with many different computers, deploying a new operating system is not an easy task. Deployment can broadly happen in two ways – upgrading and imaging. Upgrading involves the in-place upgrade of the old operating system using the new operating systems's setup program. Imaging involves laying out the image of the new operating system using file-level copy or bit-level copy (without using any setup programs)

One way that operating system deployment can occur is to have a system administrator physically visit each of the machines in the organization and

1 physically install the new operating system. Needless to say, this approach is a
2 very time consuming and intrusive task not only for the system administrator, but
3 for the user of the machine as well. In organizations with large numbers of
4 machines that are to be upgraded, this approach is not practicable.

5 There is a continuing need to provide methods and systems that automate
6 the process of deploying operating systems in the context of large hierarchies of
7 computers, such as those found in various organizations such as companies and the
8 like.

9 10 SUMMARY

11 Various embodiments provide systems and methods that can be utilized to
12 remotely manage operating system deployments. Various features can allow a
13 system administrator to plan and schedule operating system or image deployment
14 on various computers or machines throughout an organization. Planning and
15 scheduling image deployment, in accordance with the various systems and
16 methods described below, can take place in bandwidth-sensitive environments and
17 are easily scalable. In addition, at least some embodiments can provide end users
18 with flexibility to alter the behavior of the image deployment within policies that
19 are specified by the system administrator. Additional embodiments provide a rich
20 mechanism by which status reports are generated and sent to the system
21 administrator to assist them in managing/tracking the deployment. Further
22 embodiments can conduct operating system deployment *in-place*, meaning that
23 additional disk partitions are not required. Additional embodiments can be utilized
24 by third parties to assist enterprises in deploying images in third party image
25 formats.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates a high level view of an image deployment process in accordance with one embodiment.

Fig. 2 illustrates an exemplary pre-notification dialog in accordance with one embodiment.

Fig. 3 illustrates an exemplary notification dialog for user-adaptive notification, approval and postponement, in accordance with one embodiment.

Fig. 4 is a flow diagram that describes steps in a method in accordance with one embodiment.

Fig. 5 is a flow diagram that describes steps in a method in accordance with one embodiment.

Fig. 6 illustrates a multi-phase image deployment process in accordance with one embodiment.

DETAILED DESCRIPTION

Overview

Various embodiments described below provide systems and methods that can be utilized to remotely manage operating system deployments. Various features can allow a system administrator to plan and schedule operating system or image deployment on various computers or machines throughout an organization. Planning and scheduling image deployment, in accordance with the various systems and methods described below, can take place in bandwidth-sensitive environments and are easily scalable. In addition, at least some embodiments can provide end users with flexibility to alter the behavior of the image deployment

1 within policies that are specified by the system administrator. The described
2 embodiments provide a complete, highly integrated, end-to-end remote image
3 deployment environment in which the targeted machines can be completely
4 managed during operating system upgrade cycles. The solutions provided by the
5 various embodiments can greatly reduce the cost and complexities of operating
6 system deployments. Hence, advantages are achieved for organizations and
7 system administrators that utilize the inventive systems and methods, as well as
8 end users of the machines that are imaged using the techniques described herein.

9 Preliminarily, the embodiments described below leverage aspects of
10 Microsoft's Systems Management Server (SMS) 2003 architecture—which is a
11 software distribution architecture. The SMS 2003 architecture provides a
12 comprehensive solution for change and configuration management for the
13 Microsoft platform, thus enabling organizations to provide relevant software and
14 updates to users quickly and at reduced cost. A commercially available product
15 that will contain features that implement the embodiments described below is the
16 “Microsoft Systems Management Server 2003 OS Deployment Feature Pack”.

17 It is to be appreciated and understood, however, that the SMS 2003
18 architecture and environment constitutes but one exemplary paradigm in which
19 image deployment can take place. Other paradigms can be utilized without
20 departing from the spirit and scope of the claimed subject matter.

21 In the description that follows, various aspects of the embodiments will be
22 described in terms of a process or steps that can be carried out by suitably
23 programmed computers, also interchangeably referred to as “machines” throughout
24 the text. It is to be understood and appreciated that the described processes and
25 steps can be embodied, where appropriate, in the form of computer-readable

1 instructions that can reside on any suitable type of computer readable medium.
2 These instructions, when executed by one or more processors, perform the various
3 processes or steps described below. By way of example and not limitation, these
4 types of computer-readable media can comprise random access memory (RAM),
5 read only memory (ROM), programmable read only memory (PROM), electrically
6 erasable programmable read only memory (EEPROM), CD, and any other type of
7 permanent or temporary type of storage.

8 The various embodiments described below can provide advantages that
9 include (1) in-place imaging, (2) user-adaptive notification, approval, and
10 postponement, (3) image deployment over bandwidth-sensitive intermittent links,
11 and (4) rich status messaging for troubleshooting individual computer as well as
12 systematic issues, all of which are described in greater detail below.

13 14 **Exemplary Image Deployment Utilizing the SMS 2003 Architecture**

15 The process of image deployment, in accordance with the embodiments
16 described below, can be thought of as having a number of different stages which
17 include image creation, image management, image distribution, and finally image
18 deployment on the destination computers or target collection of machines.

19 To better assist the reader in understanding the overall process of image
20 deployment, particularly in the context of the SMS 2003 architecture, please refer
21 to Fig. 1 which illustrates a higher level view of the image deployment process by
22 showing the data/information flow generally at 100. The discussion that follows
23 breaks out a discussion of each of the stages under their own respective headings.
24
25

Image Creation Process

Initiating the image deployment process is the image creation process, shown generally at 102. During this process, a system administrator creates the image that is to be deployed across the various target machines. In this embodiment, the system administrator sets up a model computer with the operating system that is to be imaged. In addition, the system administrator can install all of the pertinent applications that will be used to execute on the operating system that is to be deployed.

In the context of SMS 2003, an *Image Capture Wizard* application is executed and enables the system administrator to capture a system image. By way of example and not limitation, the *Image Capture Wizard* application presents the system administrator with a number of different user interface pages that step the administrator through the image capturing process. For example, after an introductory page, a *source and destination page* can be presented to the system administrator and allows the administrator to specify the drive that will be used as a source for generating the image, the name of the image file, a network share where the image file will be stored, and the account and password that will be used to access the network share.

An *image properties page* allows the system administrator to specify image properties that will be stored inside an image file that is created. This includes information that is helpful for deploying the image to target machines at a later stage. A *completion page* allows the system administrator to complete the image capture wizard, and start the image capture process. A *progress page* can display indicia of the image capturing process, e.g. by using a progress bar. In addition, the *progress page* can also display the name of the source file being captured, data

1 already captured, and elapsed and remaining time estimates. Various other user
2 interface pages can be presented to the system administrator such as a *cancel*
3 *confirmation page* to cancel the image capture, an *error handling page* for
4 enabling the system administrator to handle errors that can occur during the image
5 capture process, and various other pages such as a *completion page* which is
6 displayed at the successful conclusion of the image capture process. Upon
7 completion of the image capture process, the image file can be compressed and
8 saved as a *network share* specified by the administrator. Part of the image capture
9 process can add the captured image to a database that is subsequently used during
10 the image management process described just below.

11 Visual examples of these user interface components are provided in the
12 related provisional application which is incorporated by reference above.

13 14 **Image Management Process**

15 Once the image has been captured from the model computer and saved as a
16 network share, the image management process can begin. In one embodiment,
17 image management can take place utilizing a *System Administrator Console* 104.
18 In the context of SMS, this console is referred to as the *SMS Administrator*
19 *Console*.

20 During the image management process, an *image package* (also referred to
21 as an “image file package”) is created that contains the image file that was created
22 during the image capture process, along with all of the necessary tools that are
23 needed by the image. An exemplary image package is shown diagrammatically at
24 106. For example, from the *Administrator Console* the system administrator can
25 create an image package that essentially contains a system image, configuration

1 information, and a *deployment environment* and its executable. The deployment
2 environment provides all of the necessary mechanisms to install the new system
3 image, as will become apparent below.

4 In one embodiment, the deployment environment comprises Microsoft's
5 WinPE™ operating system. WinPE™ is an operating system based on Microsoft
6 Windows® XP Professional that facilitates building custom solutions that speed
7 up deployment through automation. WinPE™ can run Windows setup, scripts,
8 and imaging applications, as will be appreciated by the skilled artisan.

9 Specifically, during image management, the system administrator creates
10 the image file, specifies a package name, and specifies the folder that contains the
11 source files for the package, as well as the paths and files names for a *package*
12 *definition file* and an *image deployment file*.

13 The package definition file is used to set all SMS 2003 package and SMS
14 2003 program properties except those that are overridden for image deployment
15 purposes. The image deployment file is used to set image deployment properties.
16 After the image package has been created, it can be added to a suitable database,
17 as shown, for distribution during the image distribution process.

18 19 **Image Deployment Files**

20 With respect to *image deployment files*, consider the following. In one
21 embodiment, the system administrator can specify, through an image deployment
22 file, properties that are to be migrated from pre-existing computer configurations.
23 For example, the system administrator can select various machine state properties
24 for migration (or to default to) such as computer name, user name, organization
25

1 name, product key (for anti-piracy measures), administrator password, and IP
2 address.

3 Additionally, the system administrator can specify which pieces of SMS
4 client state need to be migrated from the pre-existing client on the destination
5 computer or configured fresh on the destination computer after the image has been
6 deployed. Examples of SMS client state information include such things as the
7 SMS GUID (which identifies the machine uniquely as an SMS client), SMS client
8 assignment, and SMS client policy.

9 Additionally, the system administrator can specify various deployment
10 options for image deployment. For example, in one embodiment, the system
11 administrator can select an option that ensures that the image will not be deployed
12 again on the same computer if the same image is currently applied to the
13 computer. Alternately or additionally, the system administrator can specify an
14 option that enables the user to give their permission to have the image deployed on
15 their computer. If the user does not give their permission to have the image
16 deployed on their machine, then image deployment does not take place. Further,
17 in some embodiments, the system administrator can select an option that allows
18 the user to specify when image deployment is to take place. For example, in this
19 embodiment, the user can specify in days and/or hours. This gives the user the
20 opportunity to flexibly schedule image deployment. This aspect is explored in
21 more detail in the section below entitled "User-Adaptive Notification, Approval
22 and Postponement".

23 Additionally, the system administrator can specify an appropriate location
24 for saving user application settings and/or user data files during image
25 deployment. In case remote backup to a network location is requested, a network

1 location, and the account name/password combination needed to gain write access
2 to it can also be specified. Further, the system administrator can specify the
3 applications whose settings will be migrated to the destination computer during
4 image deployment. Additionally, the system administrator can specify the file
5 types and locations of user data that will be migrated during image deployment.

6 7 Image Distribution Process

8 Once the image has been captured and the image file package has been
9 created and managed by the system administrator, the image file package can now
10 undergo an image distribution process. One goal of the image distribution process
11 is to distribute the image so that one or more target computers can be imaged.

12 In accordance with one embodiment, image file packages are distributed to
13 one or more *distribution points (DP)*, which is diagrammatically shown at 108.
14 Distribution points can effectively serve as geographically disperse *file shares*
15 where individual clients can obtain the image file packages. Using distribution
16 points along with a multi-phased deployment, discussed below, enables image
17 deployment to function quite well in bandwidth-limited environments that have
18 intermittent links. This is particularly the case in embodiments that utilize the
19 SMS architecture for image deployment.

20 Once the image file package has been distributed to one or more
21 distribution points, the package can be *advertised* to the target or destination
22 computers. That is, the system administrator can specify a collection of computers
23 to which the image file package is to be advertised. Once the system administrator
24 has distributed the image file packages and configured an advertisement,
25

1 individual computers of the collection can receive the advertisement, which is
2 diagrammatically shown at 110.

3 4 **Image Deployment Process**

5 Image deployment can be considered to begin when advertisements are sent
6 to the target computers of the collection that is targeted for imaging. In some
7 embodiments, the system administrator can cause advertisements to be sent to the
8 target computers. The advertisements advise the users that their computer has
9 been selected for imaging a new operating system.

10 11 **User-Adaptive Notification, Approval and Postponement**

12 In one embodiment, the users of the targeted computers can be given
13 advance notification of the planned image deployment action, and can optionally
14 allow, disallow or postpone deployment of the image. In this embodiment, a
15 notification dialog, configured by the system administrator, can be displayed on
16 the target machine to notify the user that an image deployment is desired.

17 In some embodiments, the system administrator can configure the
18 notification dialog so that the user is simply advised of the planned image
19 deployment and informed when image deployment will occur, e.g. in 3 hours, 23
20 minutes. In this embodiment, the notification dialog can provide a countdown so
21 that the user is given an opportunity to complete their work prior to deployment.

22 In another embodiment, the system administrator can configure the
23 notification dialog so that the user has a more direct, proactive role in establishing
24 when, if at all, image deployment is to occur. As an example, consider Fig. 2
25

1 which shows an exemplary pre-notification dialog in accordance with one
2 embodiment generally at 200.

3 There, a dialog portion 202 contains a default message that is displayed in
4 the top part of the dialog, near the alert icon, as shown in the screenshot. This
5 message informs the user that a new operating system installation is desired. A
6 dialog portion 204 can be provided and allows the system administrator to custom
7 design a message for the user. A dialog portion 206 comprising a count down can
8 be provided, if enabled by the system administrator. In this example, dialog
9 portion 206 informs the user that if they do not respond to the notification, then
10 image deployment will occur automatically. However, if the user does respond,
11 then in some embodiments the user can be given an opportunity to postpone image
12 deployment. A dialog portion 208 can be provided and enables the user to disallow
13 the image deployment.

14 In some embodiments, a notification dialog that is configured by the system
15 administrator can be displayed for the user when an image deployment package
16 executes on the target machine. Specifically, in this embodiment a dialog box can
17 display an administrator-specified user interface to notify the user that image
18 deployment is starting up. The dialog box can be configured to allow the user to
19 postpone deployment of the image, wait for an administrator-specified countdown
20 interval, deploy the image right away, or decline the image altogether. As an
21 example, consider Fig. 3 which shows an exemplary notification dialog in
22 accordance with one embodiment generally at 300.

23 There, a dialog portion 302 comprises a default message that is displayed in
24 the top part of the dialog, near the alert icon, as shown in the figure. This dialog
25 portion can also contain a count down, if enabled by the system administrator, that

1 displays a running countdown of remaining hours and minutes until deployment.
2 A dialog portion 304 can be provided and can enable the system administrator to
3 configure a custom message for the users.

4 Dialog portion 306 comprises the portion of the dialog that enables the user
5 to proactively participate in the imaging of their computer. Specifically, an
6 “Install Now” button 308 allows the user to immediately begin image deployment.
7 An “Install Later” button 310 enables the user to either install later as scheduled,
8 or enter his or her desired postponement duration (within a maximum limit which
9 can specified by the administrator). A “Decline” button 312 enables the user to
10 cancel the image deployment.

11 Fig. 4 is a flow diagram that describes steps in a method in accordance with
12 one embodiment. The method can be implemented by software that executes on a
13 destination computer that is targeted for imaging. In some embodiments, this
14 software can comprise both a client that is associated with the old operating
15 system, and a client that is associated with the new operating system, as will
16 become apparent below.

17 Step 400 notifies a user that image deployment is desired for their particular
18 computer. Notification can take place using any suitable means. In one
19 embodiment, notification can take place by using a dialog box, such as the one
20 shown and described in connection with Fig. 3. Step 402 receives user input that
21 pertains to the image deployment. The user input can comprise any suitable type
22 of user input. In the Fig. 3 example, the user is given an opportunity to either have
23 the image deployed immediately (i.e. install now), postpone the image deployment
24 (i.e. install later), and/or decline the image deployment all together.

1 In an event that the user opts to postpone the image deployment, in at least
2 some embodiments, the user is given an opportunity to flexibly schedule the
3 deployment by selecting an appropriate time for deployment, or a time interval
4 after which deployment will occur.

5 Step 404 performs image deployment pursuant to the user's input. For
6 example, if the user opts to install the image "now", then the image will be
7 immediately installed. If the user opts to postpone image deployment, the
8 deployment will be postponed. If the user opts to decline image deployment, then
9 the image will not be deployed.

10 Assume now that the image package has been advertised and, as part of the
11 advertisement execution, image deployment now begins at 112 (Fig. 1).

12 Fig. 5 is a flow diagram that describes steps in an image deployment
13 method in accordance with one embodiment. Following the description of this
14 method, an implementation example is described to assist the reader in
15 understanding certain nuances associated with image deployment.

16 Step 500 captures and preserves various data for migration to the new
17 operating system that is to be imaged. Any suitable data can be captured and
18 preserved, e.g. data associated with the machine, the user, the client executing on
19 the user and the like. A specific example of these different types of data is given
20 below. Preservation of the data can take place by saving the data in a suitable file
21 that can be later used in a restoration process, as will become apparent below.

22 Step 502 installs a pre-installation environment and re-boots the machine in
23 the pre-installation environment. By using a pre-installation environment in
24 connection with the image deployment, the image can be deployed *in place*,
25 meaning that the new operating system can be deployed in the very same disk

1 partition that originally contained the old operating system. It is to be appreciated
2 and understood that various inventive features described herein can be utilized in
3 scenarios in which imaging is not conducted in place. There are, however,
4 advantages associated with performing the imaging in place. For example, some
5 machines may not have separate physical partitions. In these contexts, the
6 inventive image deployment techniques can flexibly accommodate such
7 machines. Another advantage of the in place imaging is that non-operating system
8 data on the disk can be preserved.

9 Step 504 deletes the old operating system from within the pre-installation
10 environment. Once the old operating system is deleted, step 506 installs the new
11 image and re-boots the machine using the new operating system. Step 508 then
12 restores the computer using captured and preserved data from step 500.

13 For purposes of the implementation example that follows, image
14 deployment can be divided into three different phases—a preparation phase (Phase
15 1), a wipe and load phase (Phase 2), and a restoration phase (Phase 3), each of
16 which is individually discussed below. These phases and their constituent parts
17 are diagrammatically illustrated in Fig. 6.

18 19 **Phase 1—Preparation Phase**

20 Phase 1 of the image deployment process effectively prepares the target or
21 destination computer for imaging. During this phase, a number of things can
22 occur. First, an initial check can be performed to ensure that the destination
23 computer is in the correct state for imaging. Specifically, part of the initial check
24 can be to ascertain whether there is enough disk space to conduct the imaging.
25 Additionally, in the SMS context, as part of the process of capturing client

1 information and machine state, checks can be performed to ascertain that the SMS
2 client site code, as well as the SMS GUID and other machine specific data is
3 assigned and/or retrievable. Next, various state information associated with the
4 destination computer and/or the user can be captured and preserved for migration
5 after imaging the new operating system. State capturing can be performed by a
6 state capture tool, such as one that is employed to capture user state data. Other
7 mechanisms or agents can be used to capture user data, machine state and client
8 state.

9 The machine state that is captured can include such things as the computer
10 name, the domain, network settings and the like. These settings can be specified
11 in a file which, for purposes of example, is referred to as the sysprep.inf file. The
12 client state that is captured typically includes information that is needed or used to
13 restore the computer during phase 3, e.g. site association or code (e.g. a site code,
14 such as one used in the SMS context, can identify which SMS Primary Site Server
15 the SMS client sends its inventory data to, and from which it receives software
16 distribution policies, advertisements and the like), client globally unique identifier
17 (GUID), associated distribution point and the like. In embodiments that utilize the
18 SMS architecture, the client state information that is captured and preserved
19 pertains to the SMS client that is executing on the destination computer. The user
20 state that is captured can include such things as the current logged-on user profile,
21 user-specific settings like Internet browser favorites list, wall paper on the user's
22 desktop, the screensaver selected by the user, and the like. The user data that is
23 captured and preserved can include folders and files that are desired to be
24 migrated, network share settings that the like. All of the information that is
25

1 captured can be stored locally on the destination computer or optionally on a
2 network location.

3 Following completion of the state/data capturing process, a pre-installation
4 environment can be installed on the destination computer. That is, before deletion
5 of the old operating system, a pre-installation environment is installed on the
6 destination computer so that the old operating system can be deleted. The pre-
7 installation environment effectively comprises a subset of an operating system that
8 is installed on the hard disk. Once installed, the destination computer is re-booted
9 into the pre-installation environment. In embodiments that utilize the SMS
10 architecture, the pre-installation environment comprises WinPE™ whose
11 operation and functionality will be understood by those of skill in the art.

12 Installation of and rebooting into the pre-installation environment
13 concludes phase 1.

14 15 **Phase 2—Wipe and Load Phase**

16 Phase 2, referred to as the wipe and load phase, effectively deletes the old
17 operating system and installs the new operating system. Specifically, once the
18 pre-installation environment has been installed and the destination computer has
19 been re-booted into the pre-installation environment, a network services
20 component of the pre-installation environment is started. This effectively enables
21 the destination computer to connect with an associated distribution point.
22 Connecting with the distribution point enables the destination computer to access
23 the necessary image file, the deployment settings and other necessary tools that are
24 utilized during the wipe and load phase. Specifically, connecting with the
25

1 associated distribution point enables the destination computer to access the image
2 packages and deploy the image file that resides at the distribution point.

3 In one embodiment, a deployment agent can use the pre-installation
4 environment to perform a disk clean up process in which the disk partition in
5 which the old operating system resides has the old operating system's associated
6 folders and files deleted. After the old operating system is effectively removed
7 from the disk partition, the new image is installed. Here, the deployment agent
8 accesses the image file, decompresses the image file and installs the files and
9 folders associated with the new operating system into the disk partition on the
10 destination computer. It is during the image installation process that the entire
11 directory structure and its contents are laid out on the disk partition.

12 After the new image is installed, the deployment agent can take steps to
13 restore the machine state. That is, recall that during phase 1, the machine state and
14 settings of the destination computer were captured and specified in a file
15 sysprep.inf. In one embodiment, the sysprep.inf file is modified to restore the
16 computer specific information. Modification of the sysprep.inf file entails, in this
17 embodiment, writing all of the machine state-specific information into the file so
18 that the information can be migrated for the newly installed image.

19 At this point in the process, the destination computer is prepared for what is
20 referred to as a "mini setup". The mini setup executes to use all of the information
21 that was captured in phase 1, to restore the new operating system. At this point,
22 however, mini setup has not been executed.

23 At the conclusion of phase 2, the destination computer is re-booted into the
24 new operating system and phase 3 begins.
25

Phase 3--Restoration

Phase 3 is the phase in which the destination computer is restored using the information that was captured during phase 1. In one embodiment, before booting into the new operating system, mini setup is executed. Then, the new operating system will be initialized and started. At this point, a *restore agent* is executed. The restore agent has several steps that it performs. First, the restore agent restores the client settings that were captured in phase 1. In embodiments that utilize the SMS architecture and infrastructure, this involves restoring the SMS client specific settings. Additionally, the restore agent restores the user data that was captured in phase 1. Additionally, the restore agent can restore any of the user state data that was captured during phase 1.

At the conclusion of phases 1, 2 and 3 the destination computer is now fully configured with the new operating system and is ready for operation.

Status Reporting

Notice in the illustration of Fig. 6 that a column entitled "Status Reporting" occurs throughout the various phases. In accordance with one embodiment, a rich collection of status reports can be developed, maintained and transmitted during the preparation phase, the wipe and load phase, and the restoration phase. For example, various status reports can be generated and sent to the system administrator to inform them on how the imaging process is progressing.

The status reporting process can track errors and events that might arise during each of the individual phases. For example, if during the preparation phase, a problem is encountered capturing the machine state, a status report can be generated and sent to the system administrator to apprise them of the situation.

1 Likewise, if a problem is encountered during installation of the pre-installation
2 environment, the system administrator can be advised of this by a status report.

3 Alternately or additionally, the status reports can be generated to track the
4 progression of the various phases. For example, a system administrator might be
5 advised through various status reports when the machine state has been
6 successfully captured, the client state has been successfully captured and the like.
7 Further, the system administrator might be advised when the disk has been
8 successfully cleaned up (phase 2) and when the machine state has been restored.
9 Advantageously, the status reporting function takes place across the operating
10 system deployment so that the system administrator has an opportunity to receive
11 a very rich collection of status reports so that they can monitor the progress of the
12 deployment across one or more machines..

13 14 **Status Reporting Using Old and New Clients**

15 One interesting feature of the status reporting embodiment, and one which
16 adds to the perceived seamlessness of the deployment, is the notion of performing
17 the status reporting using an old client that executes on the machine prior to
18 deployment and a new client that is installed as part of the deployment.
19 Specifically, and in the context of the SMS infrastructure, consider the following
20 example.

21 In the SMS context, in order to receive and process the advertisement
22 associated with the desired operating system deployment, an SMS client is
23 provided on the target machine. The SMS client, in this example, comprises the
24 software code that is familiar with the SMS architecture and deployment process.
25 As the advertisement is received and the target machine enters phase 1 (Fig. 6), the

1 SMS client (referred to in this example as the “old client”) on the target machine
2 guides the target machine through phase 1 and generates status messages that are
3 returned to the system administrator. Now, when the target machine enters phase
4 2—the wipe and load phase—not only is the old image removed from the hard
5 disk, but the old client is removed as well and a new client is installed. The new
6 client continues the imaging process and continues to generate and transmit status
7 reports for the system administrator.

8 To the system administrator, it appears as if the status reports have been
9 continually generated in a seamless manner. In reality, at least in this
10 embodiment, the status reports have been generated by the old client (prior to the
11 wipe and load phase) and the new client (after image installation).

12 13 **Third Party Image Deployment**

14 One advantageous feature of the described embodiments is that the
15 infrastructure described above can be utilized by third parties to assist enterprises
16 in deploying third party image formats. Enterprises may have spent millions of
17 dollars in capturing Operating System images in these third party image formats.
18 Yet, when these enterprises wish to deploy these operating system images in third
19 party formats, they are faced with the very same problems described in the
20 “Background” section—i.e. cost and inconvenience. By providing a
21 comprehensive set of application program interfaces (APIs), these enterprises can
22 conduct image deployment with popular image formats with a greater degree of
23 convenience and flexibility than has been afforded in the past.

24 As such, the APIs can implement the functionality that is described above
25 to enable third parties, in at least some embodiments, to allow the SMS 2003 OS

1 Deployment Feature Pack to create an image, manage the image, distribute the
2 image and deploy the image.

3 4 Conclusion

5 The various embodiments described above provide several mechanisms that
6 can perform end-to-end, remote operating system imaging of a client computing
7 device. In a typical hierarchical scenario, such as a corporate scenario, the
8 deployment can be performed across the network, even with bandwidth constraints
9 or intermittent connectivity. Multiple phases and distribution points are utilized to
10 mitigate the impact of these bandwidth constraints and/or intermittent
11 connectivity.

12 Various embodiments provide system administrators with a comprehensive,
13 complete, and up-to-date status of their deployments and enable the administrators
14 to track and, if necessary, troubleshoot the deployment. Status messages or
15 reports can also be categorized and summarized to help flush out systemic trends
16 and issues. At the same time, the end-user whose desktop is being imaged, can be
17 provided with a desirable degree of flexibility so that the deployment does not
18 interrupt or interfere with their work and can be approved or postponed as per their
19 individual needs and according to the policies specified.

20 In various embodiments, by staging image deployment in phases on a
21 network share that is accessible to the client computing device, the need for an
22 active server component that is required by most of the existing technologies that
23 perform image deployment can be eliminated.

24 In some embodiment, the actual image deployment process that takes place
25 on the targeted client computer is unique in the sense that the image can be applied

1 *in-place*, without requiring additional disk-partitions, bootable disks or CDs,
2 which is unlike any other existing image deployment technologies.

3 Although the invention has been described in language specific to structural
4 features and/or methodological steps, it is to be understood that the invention
5 defined in the appended claims is not necessarily limited to the specific features or
6 steps described. Rather, the specific features and steps are disclosed as preferred
7 forms of implementing the claimed invention.